

РОЗВ'ЯЗУВАННЯ ТВОРЧИХ ЗАДАЧ З ПРОГРАМУВАННЯ МАЙБУТНІМИ УЧИТЕЛЯМИ ІНФОРМАТИКИ

У статті досліджуються питання формулювання такого терміну, як "творча задача з програмування", наводиться авторська класифікація творчих задач з програмування, розглядаються деякі типи розв'язування таких задач.

Ключові слова: задача, творча задача, задача з програмування, творча задача з програмування, класифікація творчих задач з програмування.

Традиційні методи закріплення та перевірки знань у процесі навчання інформатики, зокрема й програмування (лабораторні роботи, тести, аудиторні та поза аудиторні модульні контрольні роботи) хороші тим, що показують, наскільки точно засвоєний матеріал, що вивчався. Однак ці методи не дозволяють виявити те, як студент навчився застосовувати різноманітні знання, використовувати їх та оперувати ними.

Використання творчих задач на заняттях з програмування значно підвищує ефективність навчального процесу. З умови задач такого типу безпосередньо не можна сказати про те, які саме знання знадобляться студентам для їх розв'язання, а тому діяльність студентів спрямована на виявлення шляхів розв'язання і підбір потрібних даних, відомостей та закономірностей. При вирішенні творчих задач можуть знадобитися різноманітні знання з інших предметів, наприклад з алгебри, геометрії, теорії ймовірностей тощо. Основною ознакою того, що студенти знаходяться у творчому процесі є відмова від традиційних підходів до інтерпретації існуючих відомостей.

До проблем розв'язування творчих задач у різних галузях зверталось багато науковців, серед них: Е. Григорова [2], А. Давиденко [3], С. Даниленко [4], К. Кноп [7], І. Лернер [9], Ю. Мурашковський [11], С. Притуляк [13] та інші. Але сам термін "творча задача" вводили лише деякі учені, наприклад І. Лернер [9]. Що ж до терміну "творча задача з програмування", то до нього звертались багато вчених, але жоден не давав конкретного означення і не наводив їх класифікацію.

Метою даної статті є виокремлення поняття "творча задача з програмування", наведення авторської класифікації таких задач, розбір розв'язку деяких задач такого типу.

Для того, щоб перейти до означення творчої задачі з програмування, необхідно спочатку звернутись до означень таких понять, як "задача", "творча задача", "задача з програмування".

Термін "задача" у тлумачному словнику російської мови трактується у різних значеннях: 1) у загальному – те, що потребує виконання, рішення; 2) у математичному – вправа, яка виконується за допомогою умовиводу, обчислення; 3) у науковому – складне питання, проблема, що вимагають дослідження і розв'язання [12]. Аналогічне розуміння даного поняття ми знаходимо і у тлумачному словнику Ушакова [16].

До терміну "задача з програмування" зверталось чимало вчених таких, як Д. Златопольський [6], С. Окулов [5], Ф. Меньшиков [10], А. Юркін [17] та інші. Ми будемо вважати, що *задача з програмування* – це така задача, яка передбачає пошук алгоритму рішення задачі засобами деякої мови програмування.

До терміну "творча задача" в різних галузях зверталось багато науковців, які наведені вище. Але дане поняття вводили лише деякі вчені, наприклад: "Творчою, – пише І. Лернер, – вважається задача, дії по розв'язуванні якої не детермінуються або не повністю (неоднозначно) детермінуються якимись прописами, тобто якщо розв'язуючому невідомий алгоритм розв'язання й необхідно здійснити пошук, кроки якого наперед не дані" [9, с. 81].

Підсумовуючи вищесказане, ми будемо вважати, що *творча задача з програмування* – це така задача, що передбачає пошук та побудову алгоритму її розв'язування з використанням існуючих методів, з подальшою реалізацією певною мовою програмування, у процесі чого учні та студенти активно засвоюють нові знання, опановують уміння та навички, розвивають абстрактне та логічне мислення, власні творчі здібності, пізнавальний інтерес.

Запропонуємо авторську класифікацію творчих задач з програмування, яку показано на рис. 1, за якою такі задачі розподіляються на три блоки: за розділами вивчення (передбачається розв'язування задач за основними темами вивчення дисципліни "Програмування"), за необхідними знаннями в інших галузях (зроблено акцент на задачах, при розв'язуванні яких мають використовуватися базові знання з відповідних предметних галузей), за рівнем складності (акцентується увага на задачах різних рівнів складності).

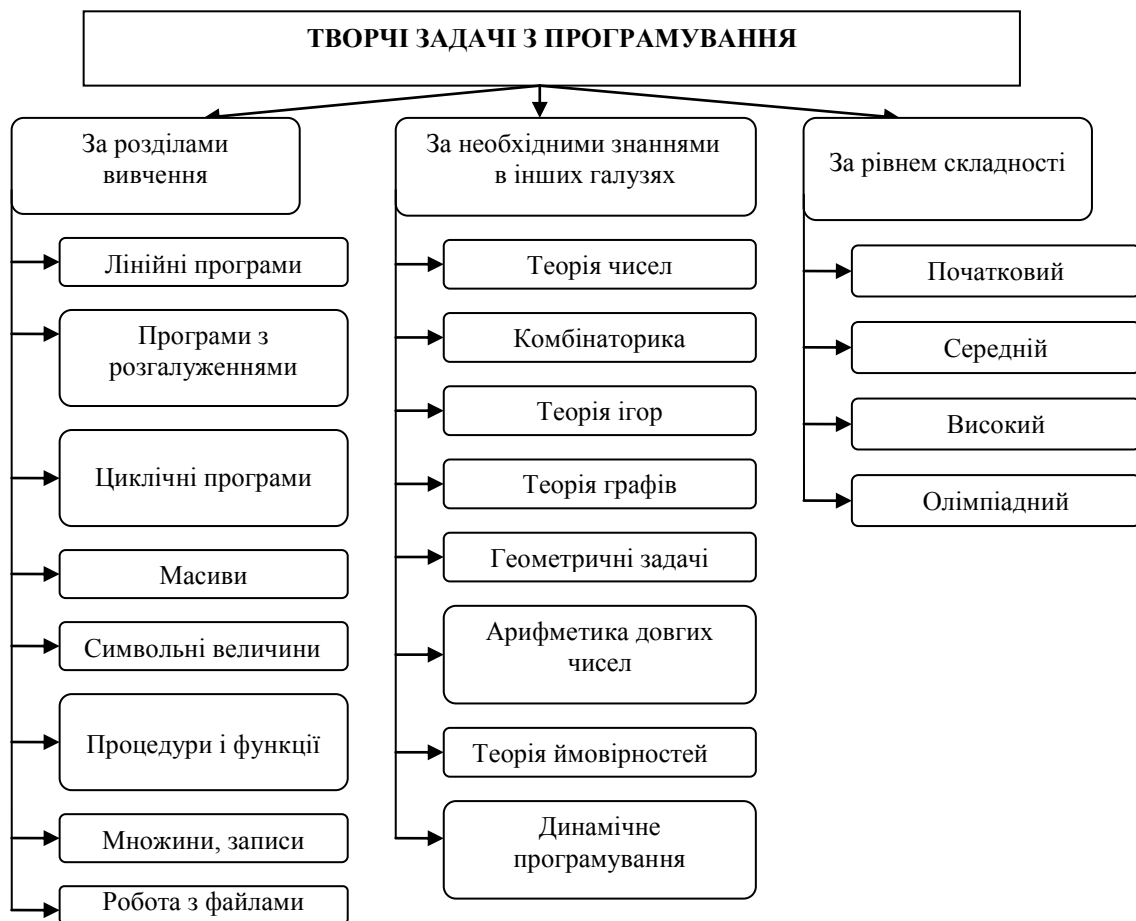


Рис. 1. Класифікація творчих задач з програмування

Наведемо приклади задач кожної класифікації до деяких видів із поясненням до розв’язання. У класифікації за *розділами вивчення* розглянемо такі види:

1. Лінійні програми

Задача 1. Дано два цілих числа a та b . Якщо a ділиться на b або b ділиться на a , то вивести на екран 1, інакше – будь-яке інше число. Умовний оператор та оператори циклів не використовувати [5, с. 17].

Розв’язання. Якщо a ділиться на b (або b на a), то це означає що при діленні в остачі буде 0. А будь-яке число в добутку з 0 дасть 0, а додавши 1 – отримаємо 1. В іншому випадку буде будь-яке інше число, крім 1.

Підказка: у розв’язку має бути оператор присвоєння, права частина якого має вигляд:

$$(a \bmod b) * (b \bmod a) + 1.$$

Як видно на даному прикладі, при розв’язанні такого плану задач студентами викладач забезпечує творчість та дискусії з однокласниками та викладачем у пошуку розв’язку задачі, необхідність пояснювати власну відповідь та розмірковувати.

2. Циклічні програми

Задача 2. Скласти програму для знаходження площі фігури методом трапецій, обмеженої кривими: $y=x^2$ та $y=x^4$ [1, с. 45].

Розв’язання. Побудувавши графік, бачимо: отримана фігура симетрична відносно осі OY (див. рис. 2). Тому шукаємо площу фігури, яка знаходиться в I четверті і результат помножимо на 2. Дана фігура проектується на вісь OX у відрізок $[0; 1]$, тому межі інтегрування: $a=0$, $b=1$. Площа фігури, обмеженої лініями, буде обчислюватись як інтеграл, де функція $f(x)=x^2-x^4$: $S = 2 \int_0^1 (x^2 - x^4) dx$.

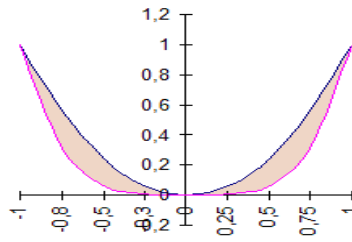


Рис. 2. Графічне зображення розв'язку задачі 2

Формула знаходження площі фігури методом трапеції має вигляд [1, с. 50]:

$$\int_a^b f(x)dx \approx \frac{b-a}{2n} \cdot \left(f(a) + f(b) + 2 \sum_{k=1}^{n-1} x_k \right).$$

Отже, відповідна програма матиме вигляд:

```
var a, b, x, s : real;
    i : integer;
begin
    a:=0; b:=1; s:=0;
    for i:=1 to 10 do
        begin
            x:=a+i/10;
            s:=s+(sqr(x)-sqr(sqr(x)));
        end;
    s:=2*(b-a)/20*(sqr(a)-sqr(sqr(a))+sqr(b)-sqr(sqr(b))+2*s);
    writeln('s=',s:10:3);
end.
```

3. Процедури та функції

Задача 3. Визначте спільні дільники двох натуральних чисел [14, с. 97].

Розв'язання. Шукаємо спочатку найменше серед двох чисел. Потім задаємо цикл від 1 до найменшого серед двох заданих чисел. Спільний дільник буде визначатись таким критерієм: якщо обидва числа при діленні на змінну, що проходить цикл, в остачі дає нуль, то це число є спільним дільником двом даним числам.

```
var a, b, c, min : integer;
{опис. функцію, яка буде визн. чи обидва числа діляться націло на параметр}
function s(a1,b1,c1:integer):integer;
begin
    if (a1 mod c1=0) and (b1 mod c1=0) then s:=1
    else s:=0;
end;
begin
    writeln('Введіть два числа');    readln(a,b);
    if a<b then min:=a {визначаємо найменше серед двох введених чисел}
    else min:=b;
    for c:=1 to min do {задаємо цикл для знаходження усіх спільних дільників}
        if s(a,b,c)=1 then write(c,' ');
end.
```

Із класифікації за необхідними знаннями в інших галузях розглянемо:

1. Геометричні задачі

Задача 4. N-кутник задано координатами своїх вершин, які вводяться по черзі за годинниковою стрілкою. Обчислити його площу, використовуючи векторний добуток [1, с. 45].

Розв'язання. Розіб'ємо многокутник на трикутники, як показано на рис. 3. Площа кожного такого трикутника буде рівна векторному добутку, поділеному на 2 [1, с. 49]). Знайшовши всі такі площі та додавши їх, отримаємо площу многокутника. Для наочності розглянемо площу одного такого трикутника. Нехай три вершини такого трикутника (на рис. 3 відповідно M_1 , M_2 і M_3) мають попарно координати x_1 , y_1 , x_2 , y_2 , x_3 , y_3 . Подамо трикутник як такий, що утворений двома векторами, які виходять з однієї точки:

$a=(a_x;a_y;a_z)$, $b=(b_x;b_y;b_z)$, де $a_x=x_2-x_1$; $a_y=y_2-y_1$; $a_z=0$; $b_x=x_3-x_1$; $b_y=y_3-y_1$; $b_z=0$. Відомо, що площа трикутника обчислюється за формулою [1, с. 49]:

$$S = \frac{1}{2} \sqrt{\begin{vmatrix} a_y & a_z \\ b_y & b_z \end{vmatrix}^2 + \begin{vmatrix} a_x & a_z \\ b_x & b_z \end{vmatrix}^2 + \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix}^2}$$

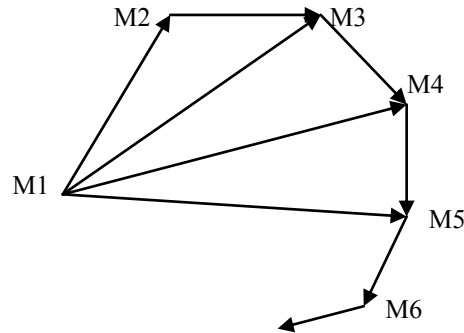


Рис.3. Многокутник поділений на трикутники

Підставляючи відповідні значення, отримаємо:

$$S = \frac{1}{2} \sqrt{\begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix}^2} = \frac{1}{2} |a_x b_y - b_x a_y|.$$

Потім для кожного наступного трикутника першу вершину (рис. 3) залишаємо без змін, другу беремо як третю вершину попереднього трикутника, а третю вершину нового трикутника вводимо знову. Обчислюємо площу для нового трикутника і так далі до останньої вершини.

Відповідна програма реалізації описаного алгоритму матиме вигляд:

```
var n, i : integer;
    x1,y1,x2,y2,x3,y3, s, ax,ay,bx,by : real;
begin
    write('Введіть n '); readln(n);
    write('Введіть попарно координати '); readln(x1,y1,x2,y2,x3,y3);
    s:=0; {знаходимо площу першого трикутника}
    ax:=x2-x1; ay:=y2-y1; bx:=x3-x1; by:=y3-y1;
    s:=s+abs(ax*by-ay*bx)/2;
    if n>=4 then {якщо вершин більше 3, то}
    for i:=4 to n do {починаючи з 4-ої вершини: першу вершину}
    begin {залишаємо без змін, 2→3, а кожну наступну}
        x2:=x3; y2:=y3; {зчитуємо як третю, і для кожної шукаємо свою}
        readln(x3,y3); {площу, додаємо до попередньої}
        ax:=x2-x1; ay:=y2-y1; bx:=x3-x1; by:=y3-y1;
        s:=s+abs(ax*by-ay*bx)/2;
    end;
    writeln('s=',s:5:2);
end.
```

2. Теорія чисел

Задача 5. Для n заданих чисел обчислити НСД за алгоритмом Евкліда.

Задача 6. Для двох заданих чисел перевірити чи є вони взаємпростими.

Таких задач досить багато, усі їх неможливо перерахувати. Задачі такого типу розв'язуються за допомогою відомих властивостей, означень з курсу математики.

Наприклад, натуральне число називається *простим*, якщо воно більше за 1 та ділиться тільки на 1 і на саме себе [15] (число 1 не є простим). Тобто, щоб з'ясувати чи є число простим, достатньо задати цикл від 1 до самого числа, і перевірити скільки чисел з даного інтервалу будуть дільниками даного числа. Якщо таких чисел 2 – то число просте, інакше – воно не є простим, а є або 1, або складеним. Наведемо фрагмент алгоритму:

```
....k:=0;
for i:=1 to n do
    if n mod i = 0 then k:=k+1;           {шукаємо k-ть дільників}
if k=2 then writeln('Число просте') else writeln('Число не є просте');
...
```

Щодо останньої класифікації, за *рівнем складності*, то досить великий вибір задач даних типів є на розробленому нами сайті www.e-olimp.com [8], де було розроблено спеціальну методичну сторінку розподілу задач за рівнями складності, яка називається Курс олімпійця [8]. Перейшовши за посиланням на відповідний рівень складності, відкриється перелік задач, які пропонуються авторами.

Отже, при розв'язуванні творчих задач з програмування у студентів формуються уміння та навички розв'язування творчих задач, що сприяє їх розумовому розвитку, а також розвиваються такі мислительні операції, як класифікація та систематизація, при розв'язанні такого плану задач забезпечується творчість та дискусії у пошуку рішення, необхідність пояснювати власну відповідь та розмірковувати.

Використані джерела

1. Вакалюк Т. А. Структурне програмування мовою Pascal (лабораторний практикум). Навчальний посібник для студентів фізико-математичного факультету. – Вид. 2-ге / Тетяна Анатоліївна Вакалюк, Сергій Станіславович Жуковський. – Житомир : Вид-во ЖДУ, 2010. – 124 с.
2. Григорова Е. Н. Использование метода непрерывного усовершенствования для формирования умений решать творческие задачи / Е. Н. Григорова // Пробл. инж.-пед. освіти. – 2008. – № 21. – С. 141–147.
3. Давиденко А. А. Творчі задачі з фізики / А. А. Давиденко // Вісник Житомирського педагогічного університету. – 2004. – № 14. – С. 101–104.
4. Даниленко С. В. Использование творческих задач по информатике для формирования у будущих учителей информатики готовности к профессиональной деятельности : автореф. дис. на соискание науч. степени канд. пед. наук : спец. 13.00.02 "Теория и методика обучения и воспитания (информатика)" / С. В. Даниленко. – Москва, 2010. – 24 с.
5. Задачи по программированию / [С.М. Окулов, Т.В. Ашихмина, Н.А. Бушмелева и др.]. – [Под ред. С.М. Окулова]. – М. : БИНОМ Лаборатория знаний, 2006. – 820 с.
6. Златопольский Д. М. Сборник задач по программированию / Д. М. Златопольский. – [2-е изд., перераб. и доп.]. – СПб. : БХВ-Петербург, 2007. – 240 с.
7. Кноп К. Творческие задачи [Электронный ресурс] / К. Кноп. – Режим доступа : URL : <http://offline.computerra.ru/1998/237/1143>. – Название с экрана.
8. Курс олімпійця [Електронний ресурс]. – Режим доступа: URL : <http://www.e-olimp.com/ua/articles/group-14>. – Назва з екрана.
9. Лернер И. Я. Дидактические основы методов обучения / Исаак Яковлевич Лернер. – М. : Педагогика, 1981. – 186 с.
10. Меньшиков Ф. В. Олимпиадные задачи по программированию / Ф. В. Меньшиков. – СПб. : Питер, 2006. – 315 с.
11. Мурашковский Ю. С. Алгоритм синтеза творческих задач [Электронный ресурс] / Ю. С. Мурашковский. – Режим доступа : URL : <http://www.temm.ru/ru/section.php?docId=3954>. – Название с экрана.
12. Ожегов С. И. Толковый словарь русского языка (онлайн версия) [Электронный ресурс] / С. И. Ожегов, Н. Ю. Шведова. – Режим доступа : URL : <http://www.classes.ru/all-russian/russian-dictionary-Ozhegov-term-8665.htm>. – Название с экрана.
13. Притуляк С. П. Роль творческих задач в развитии познавательной активности [Электронный ресурс] / С. П. Притуляк. – Режим доступа : URL : <http://bio.1september.ru/article.php?ID=200300108>. – Название с экрана.
14. Программирование на языке Pascal: задачник / [Под. ред. Усковой О.Ф.] – СПб: Питер, 2003. – 336 с.
15. Просте число [Электронный ресурс]. – Режим доступа : URL : http://cybportal.univ.kiev.ua/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE. – Название с экрана.
16. Толковый словарь Ушакова [Электронный ресурс] – Режим доступа : URL : <http://www.slovo-pedia.com/3/199/786542.html>. – Название с экрана.
17. Юркин А. Г. Задачник по программированию. Учебное пособие / А. Г. Юркин. – СПб. : Питер, 2002. – 192 с.

Vakalyuk T.A.

**SOLUTION CREATIVE PROGRAMMING TASKS
OF FUTURE TEACHERS INFORMATICS**

The article deals with the question wording such term as "creative programming tasks", the author provides a classification of creative programming tasks, discusses some types of solving such problems.

Key words: *job is a creative task, task programming, creative programming tasks, classification tasks with creative programming.*

Стаття надійшла до редакції 08.07.2013 р.